Emerging Media Interface Design
Emilie Whitewolf
Dr. Jeff Ritchie

Emerging Media FINAL Project:

Design Document

Use Case:

Museum Kiosk

User Goals:

Learn something new in a short, fun, and engaging way.

Solution:

Gesture-controlled Pepper's Ghost with interactive 3D models.

Materials:

- ➔ Roll of mylar *(From home)*
- ➔ MDF *(Makerspace)*
- ➔ Mirror *(From home)*
- ➔ Cardboard *(Makerspace)*

- ➔ Black Paint *(From home)*
- ➔ Wood glue *(Makerspace)*
- ➔ Hot glue *(Makerspace)*
- ➔ Screws *(From home)*

- ➔ Sandpaper *(From home)*
- ➔ Jigsaw *(From home)*
- ➔ Screw gun *(From home)*

- ➔ Epson projector & Power Supply *(DCOM studio)*
- ➔ UltraLeap II & Cable *(Ritchie's Office)*
- ➔ HDMI Cable *(From home)*
- ➔ Macbook *(From home)*
- ➔ Power Supply *(From home)*

Build Notes:

I used about 5 sheets of MDF to build this Pepper's Ghost Structure. The structure holds the reflective illusion surface, the projector light source, and the redirecting mirror. The reflective illusion surface is made of an MDF frame, with mylar as the reflective material and black Gaff tape to create tension in the mylar, which creates a smoother surface. I used an Epson projector as the light source, which I connected to my computer to show the content. I also connected my computer to an Ultraleap 2 Motion Tracker to observe the user's gestures and control the content.

Emerging Media Interface Design
Emilie Whitewolf
Dr. Jeff Ritchie

Content Notes:

To create the content, I used two free guitar assets--one acoustic and one electric--from the Unity Asset Store and two free guitar sound effects--also acoustic and electric--from Freesound.org. I brought these assets and sounds in the Unity Game Engine, where I used the skills I learned in this class to attach sounds, C# scripts, and orient the assets. To help with the scripts, I used ChatGPT as a starting point and edited the code myself whenever needed.

To reduce the amount of coding and consequential face palming, I decided to limit myself to the Ultraleap widgets for interactions. Specifically, I used the Pointer widget, which has a few simple mouse applications. As a result, users can interact with the content in three ways, each with an associated gesture or combination of gestures.

The first way is a simple pinch, in which the user closes and opens their pointer finger and index finger. The gesture starts when the user closes their fingers and ends when the user opens their fingers. This pinch, or click, prompts the kiosk to play a sound, either acoustic or electric, based on the visible guitar.

The second way is a double pinch, in which the user quickly closes and opens their pointer finger and index finger twice in a row. The gesture starts when the user closes their fingers for the first time and ends when they open their fingers for the second time. This double pinch, or double click, prompts the kiosk to switch between the electric and acoustic guitars.

The last way is a pinch and raise, in which the user closes their pointer and index finger and moves their hand up or downwards. The gesture begins when the user closes their fingers and ends when the user opens their fingers. This pinch and raise, or click and scroll, allows the user to rotate the guitar on the Y axis, giving them a full view of the model.

Reflection:

Compared to my last Pepper's Ghost, this one was bigger, used a different light source, and had more interactability, which means I achieved all three of my goals to improve the base project. My first one was about the size of a microwave, used a small monitor placed above the reflective surface as the light source, and could only be controlled by pinching. This new one, however, is slightly larger than a minifridge, uses a projector and mirror placed below the reflective surface as the light source, and can be controlled by pinching, double pinching, and pinch and raising.

Overall, this project was challenging and fun. If I could go back and do it again, I would have experimented with angles and box sizes to make a brighter, bigger, and more impressive illusion. I also would have spent more time on the interactable content in an attempt to use more than just clicking and scrolling gestures. I learned a lot more about C# throughout the process than I meant to, but now I have a firmer grasp on Unity and its capabilities.